PolylineM Route Building Process for UDOT LRS Geometry Layer

Contributed by Bert Granberg
10, Nov. 2009
Last Updated 11, Nov. 2009

Live Document, Last Updated 11/11/09 11:52am

AGRC and UDOT are collaborating on the maintenance of a statewide roads GIS layer from which the route features can be derived and calibrated.

The steps below describe a process of building and calibrating polylineM route features from the statewide road feature dataset maintained in Utah's State Geographic Information Database. This process can be used at anytime by UDOT staff using the road features in AGRC's editing database (UTRANS) and will also be done periodically as part of AGRC's bimonthly scheduled releases of roads related data in the SGID public-facing database (SGID93) and related shapefile and file geodatabase products on the SGID ftp site.
 - 1. Data Preparation - Streets. Features in the street centerline feature class (based on SGID93.Transportation.Roads) must be attributes with the proper UDOT route name attribute (field=DOT_RTNAME, example value=0015P for Interstate 15) and the route part attributes (field=DOT_RTPART, example value=1 for all single part routes).

 - 2. Data Preparation - Calibration End Points. Most routes are composed of a single, connected polyline part. However, there are a fair number of route that must be built from two or more component parts. This must be done for 3 cases: A) concurrent routes that run concurrently with other higher order routes (examples: US89 between Bountiful and Farmington, I-84 between Tremonten and Ogden); B) broken routes, where the route stops in one location and then starts up again in another location (example: SR-30 in Box Elder, Cache, and Rich counties; and C) branching or looping routes (examples: SR-190 in Big Cottonwood includes both a branch and a loop). Each route part must have two end point features stored in the UTRANS.TRANSADMIN.UDOTRoutePartEndPoints feature class that carry the route name, direction, and part number (in the LABEL field, a concatenation of RT_NAME, RT_DIR, and RT_PART) and REF_VALUE (milepost value at start and end, respectively).

 - 3. Build Simple Calibrated Routes. The first step in route building is to build simply calibrated routes. This process, described in the post VBA: Build Simple Calibrated Routes aggregates the centerline features into a single route feature for each route part, stored in a file-geodatabase.

 - 4. Merge Route Parts into a Single Route Feature. The code to do this is a function called improvedRoutePartMerge()process which is found at the bottom of the VBA: Build Simple Calibrated Routes post. This will result in a single route feature (which may be composed of several ordered parts).

 - 5. QC Simple Route Features. Rules that must be enforced include
 - M Coordinates must ordered with the direction of the route feature (0 must be at the start point) and must be increasing across the entire route feature. This means that correct ordering of the parts is essential and is why this is done with a script instead of with the ArcMap merge functionality.
 - M coordinates can be repeated where vertices are within the resolution (5 feet) of one another but more than one 0.000 m coordinate is not allowed in the Oracle environment
 - Other requirements here...

 - 6. Apply Intermediate Calibration Points to Route. In an ideal world, the mileposts are the monuments for the linear referencing system and a high-precision location would be available for all mileposts in order to provide for a dense, precise mid-route calibration. Until they are all collected, a set of midpoint calibration points can be employed from the UTRANS.TRANSADMIN.UDOTRouteReferencePoints feature class. This process is described in the post ArcMap .Cal Script: Refine Route Calibration with Milepost Point where a .cal script is currently utilized (needs to be slightly expanded to work across multiple routes.

 - 7. QC Intermediate Point Calibrated Route Features. Same rules under step 5 above need to be assessed.

 - 8. Load Routes into SDE Feature Classes. Archive the existing route feature class before loading the new routes. To load the new routes, use the 'Delete Features' command in the ArcGIS command line interface to truncate/empty existing routes and then, in ArcCatalog, right click on the newly emptied SDE route feature class and select the 'Load' context menu option use the simple data loader to import the new routes into SDE. Do this for route feature classes in both the UDOT (UDOT.UDOTADMIN.AGRC_Routes_IPCalibrated)and SGID93 (SGID93.TRANSPORTATION.UDOTRoutes_CalibratedIP) databases. IP stands for intermediate point calibration. Shapefile and FileGDB files on the SGID ftp site should also be updated.

 - 9. Derive and Update Interpolated Milepost Points. Until a complete, high-precision set of mileposts are available, an approximation of milepost locations can be derived using the process described in this post. VBA: Generate Milepost Locations From PolylineM Routes. The mileposts generated throgh this process currently get loaded into SGID93.Transportation.UDOTMileposts_Approx. Shapefile and FileGDB files on the SGID ftp site should also be updated.

 - 10. Recalculate Milepost To/From Ranges on Centerline Road Features (AGRC). Use the code in this post, VBA: Transfer Milepost M Coordinates From Route To Features, to update the fields DOT_F_MP and DOT_T_MP in UTRANS.TRANSADMIN.StatewideStreets.